

# Flexible and Modular Support for Multicast Rate Adaptation in WLANs

Nicolò FACCHI<sup>1</sup>, Pierluigi GALLO<sup>3</sup>, Francesco GRINGOLI<sup>1</sup>, Stefano PARIS<sup>2</sup>,  
Ilenia TINNIRELLO<sup>3</sup> and Antonio CAPONE<sup>4</sup>

<sup>1</sup>CNIT / Università di Brescia, DII, Brescia, Italy

Emails: {nicolo.facchi, francesco.gringoli}@ing.unibs.it

<sup>2</sup>LIPADE, Paris Descartes University, France

Email: stefano.paris@parisdescartes.fr

<sup>3</sup>CNIT / Università di Palermo, DEIM, Palermo, Italy

Email: {pierluigi.gallo, ilenia.tinnirello}@tti.unipa.it

<sup>4</sup>Politecnico di Milano, DEIB, Milano, Italy

Email: capone@elet.polimi.it

## Abstract:

The flexibility and virtualization capabilities provided by wireless cards have received significant attention as a means to reduce development costs. In this paper we present a modular architecture that exploits the features provided by emerging PHY and MAC implementations to rapidly develop new rate adaptation algorithms for multicast transmission in wireless LANs. We validate our solution by developing three rate adaptation algorithms that use an innovative sensing mechanism to evaluate the frame reception correlation of the members of the multicast group.

The experimental results obtained on a real-life testbed show that our solutions permit to increase the performance of multicast transmissions up to 3x and 5x in terms of throughput and delay with respect to the standard fixed rate approach, reducing losses at the same time.

**Keywords:** Multicast Rate Adaptation, Multimedia Communication, Wireless MAC Processor, Wireless LAN.

## 1. Introduction

The evolution of 802.11 technology [1] is making wireless networking extremely popular, thanks to general low cost and competitive performance. Even if packet delivery is not as reliable as in the wire, packet retransmission mechanisms together with rate adaptation mitigate signal propagation issues to such an extent that users accepted this access technology also for receiving multimedia contents, e.g., on smartphones, tablets and Internet enabled tv boxes. Although experience may be really pleasant when watching end to end streamed content, things change when multicast delivery is involved: as there are no feedbacks from receivers at the access layer, both rate adaptation and retransmissions have not been implemented and Access Points select the slower data rate for multicast transmissions. This kills applications that could potentially fit well the broadcast nature of the radio channel.

Though IEEE is addressing the retransmission problem [2], in this paper we propose a rate adaptation technique that tunes the transmission rate of a multicast transmission dynamically without requiring any change or amendment to the access protocol. In general, rate adaptation needs two main blocks: the low-level *feedback protocol*, that collects frame delivery information from receivers; and the high-level *rate adaptation algorithm* that selects the best transmission rate according to such information. We

resort on the flexibility provided by FLAVIA architecture [3] to implement the former; in particular, we introduce a feedback mechanism that collects delivery statistics for computing the reception correlation of wireless links of the multicast group. Then we present three algorithms that works on top of the feedback protocol and that address respectively the minimization of the delay and the frame losses and the maximization of the throughput and we compare them to the standard, fixed rate, approach.

The paper is structured as follows. Section 2. describes related work. Section 3. introduces the FLAVIA architecture and the protocol used to collect channel quality information. Section 4. describes the rate adaption algorithms, and Section 5. illustrates experimental results obtained on a real-life testbed. Section 6. concludes the paper.

## 2. Related Work

The availability of flexible PHY supporting different transmission modes (single/multi antennas, short/long preambles, single/double bandwidth, etc.) is pushing the development of rate adaptation algorithms. The key issue to solve is the design of an effective feedback mechanism for assessing the channel quality towards each receiver and opportunisticly select the PHY mode and the corresponding data rate as a tradeoff between modulation robustness and air-time. Multicast scenario increases complexity as the transmitter must handle multiple feedbacks simultaneously.

Current solutions consider both open-loop [4, 5] and closed-loop [6, 7, 8, 9] approaches based on different physical parameters, such as SNR, BER or FER, related to channel quality. Apart from the availability of the relevant PHY signals, in many cases these schemes require to change the *frame formats* (coding BER measures performed at the receiver side in the ACK frame [7]), the *channel access operations* (using RTS/CTS or basic access according to the frame loss events [5]) or the *frame handshake sequences* (using a variable CTS transmission rate as a parameter robust to collisions for coding the channel quality [6]). As no commercial card supports such customizations, most of these schemes have been only simulated or validated with simplified implementations over open-source drivers and SDR platforms.

## 3. Rate Adaptation for Multicast Transmissions

In this section, we first describe the *feedback protocol* and then we illustrate its integration in the FLAVIA architecture.

### 3.1 Feedback Protocol

As shown in Figure 1, we divide air-time into two intervals to form a temporal *super-frame* composed of a *Transmission period* and a *Polling period*. The super-frame starts always with a transmission period that includes  $N$  data frames transmitted by the AP. The polling period concludes the super-frame and is coordinated by the AP for collecting feedbacks. Independently of the period, all nodes in the BSS adopt always a standard DCF access algorithm. During the transmission period of the  $e$ -th super-frame<sup>1</sup> the AP transmits frame  $f_i$ ,  $(e-1)N \leq i < eN$  at the optimal rate<sup>2</sup>  $r^i = r_b$  if the sequence counter  $i$  satisfies the condition  $i \bmod \lfloor \gamma N \rfloor \neq 0$ . For the remaining  $\lfloor \gamma N \rfloor$

<sup>1</sup>We count super-frames starting from  $e = 1$

<sup>2</sup>In this paper,  $r^i$  refers to the rate used to transmit the frame whose sequence number is  $i$ ; instead,  $r_j$  refers to the transmission rate of index  $j$  (i.e., considering 802.11g,  $r_0$  is 6Mbps and  $r_7$  is 54Mbps).

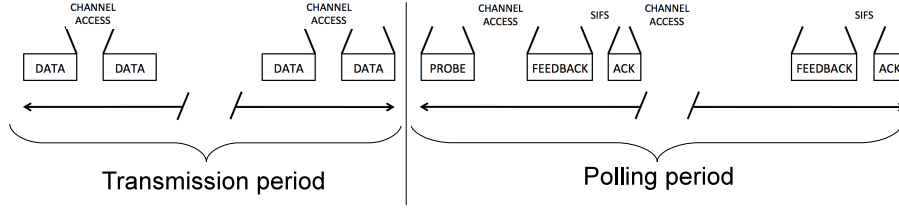


Figure 1: Channel access inside the Super-frame: standard DCF is used.

frames (called *look around frames*), whose number depends on the parameter  $\gamma \in (0, 1)$ , the AP selects a rate  $r^i = r_l \neq r_b$  according to the algorithm described in Section 4.. Using this approach, the AP can continuously evaluate the channel quality experienced with data rates other than the optimal one.

During the polling period the AP coordinates the feedback protocol illustrated in Figure 2. Specifically, after transmitting a multicast *poll* frame, the AP waits the corresponding feedbacks from all stations. If some feedback is lost, the AP restarts the poll after a short timeout, for a maximum of  $n_a$  attempts. To trade off between accuracy and overhead, in the experiments, we fixed  $n_a = 7$  and we set the timeout considering the propagation delays and the transmission time of polls and replies at the lowest bit-rate (i.e.,  $r = 6Mb/s$ ).

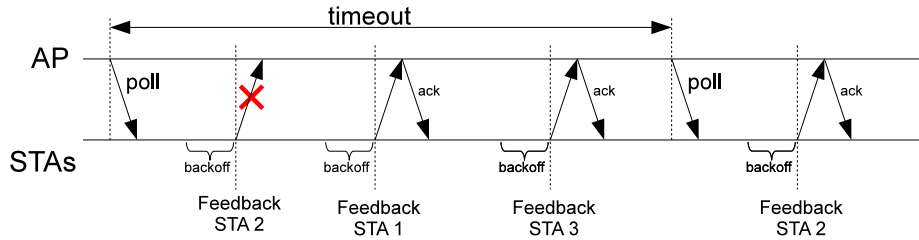


Figure 2: Feedback protocol. In the example we assume the multicast group is composed of 3 STAs.

Figure 3 shows the feedback from station  $s$  during the  $e$ -th super-frame, made of i) the sequence number of the last received multicast frame  $s_s(e)$ , and ii) an  $N$  bits bitmap  $b_s(e)$ , whose  $n^{th}$  bit represents the reception ( $b_{s,n}(e) = 1$ ) or the loss ( $b_{s,n}(e) = 0$ ) of the frame identified by sequence number  $\left(i = \lfloor \frac{s_s(e)-1}{N} \rfloor N + n + 1\right) \in [(e-1)N + 1, eN]$ .

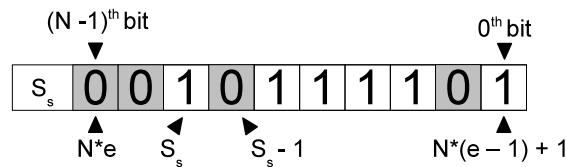


Figure 3: Feedback frame sent by stations  $s$  during super-frame  $e$ .

As explained in Section 4., all received feedbacks are transferred to the *rate adaptation algorithm* that selects the best transmission rate for the next transmission period.

### 3.2 Feedback Protocol over the WMP

High flexibility at the MAC layer is a key requirement to fully exploit the powerfulness of emerging PHYs (e.g. 802.11n/ac/ad) in different network topologies and application

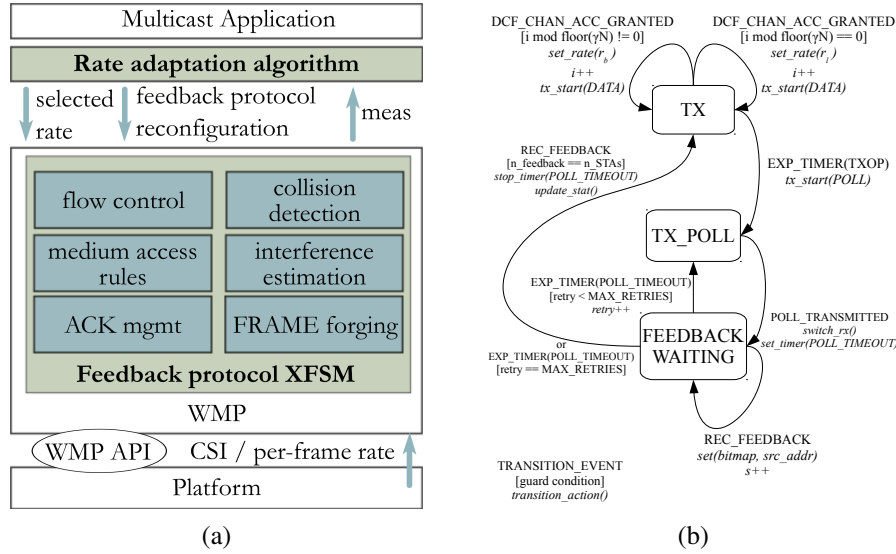


Figure 4: Example of rate adaptation modularization (a) and low-level operations for channel estimation (b).

scenarios, such as the multicast delivery considered in this paper. Nevertheless, such flexibility, which enables deep customization of probing, signaling, feedback generation and collection, is not available on any commercial device.

To counter this issue, we analyzed the possibility to implement rate adaptation for multicast transmissions within the FLAVIA architecture to fully exploit the flexibility of the Wireless MAC Processor (WMP), recently proposed in [3], which enables the reconfiguration of the wireless stack over *closed* devices. This architecture exposes a set of API and defines the behavior of MAC protocols (state transitions, triggering signals and configuration registers) in terms of hardware agnostic state machine descriptions. WMP, which is a generic state machine executor, executes these state machines<sup>3</sup>.

The WMP API already support the definition of a wide range of access solutions including TDMA, multi-channel, piggybacked ACKs: we verified in this work that simple extensions can code the low-level operations of any rate adaption scheme. As shown in Figure 4-(a), the adaptation algorithm filters the information provided by the feedback protocol and performs some actions (including the reconfiguration of the feedback protocol) according to the filtering results. Filters and decisions can work on loose time-scales (usually corresponding to the transmission of multiple frames) which are less critical than the medium access timings and can be implemented on top of the MAC state machine. Conversely, the feedback protocol requires to: i) forge new probing or signaling frames, ii) add new header fields; iii) change the medium access rules for probing or signaling frames, iv) change the handshake policies for distinguishing collisions and channel errors, v) introduce priority mechanisms for data frames to be used as probes. The WMP API supports all these requirements.

Figure 4-(b) illustrates the state machine that describes the protocol proposed in Section 3.1: the Access Point sends frames at the selected rate ( $r_b$  or  $r_l$ ) during the transmission period (TX state) and then it coordinates the polling period for collecting delivery information (FEEDBACK WAITING state) sent by receivers in response to a probe frame (TX POLL state).

<sup>3</sup>In our case WMP runs over commercial ultra-cheap 802.11 cards made by Broadcom

#### 4. Rate Adaptation for Multicast Transmissions

Hereafter, we describe the *rate adaptation algorithm* that runs on top of the WMP which executes the proposed feedback protocol.

At the end of a polling period the rate adaptation algorithm analyzes the feedbacks collected by the feedback protocol. Since stations send feedback frames after receiving the same poll, information collected during the polling period should report about the same set of transmitted data frames, unless some stations did not receive anything during the last transmission period. The algorithm can easily detect these stations, because their feedbacks include a sequence number  $s_s(e)$  lower than that of the first multicast frame transmitted in the transmission period  $e$ : in this case the algorithm assumes these stations left the multicast group and discards their bitmaps.

At this point the algorithm computes the logical AND of the remaining bitmaps: the result represents the frames received jointly (bit set to 1) by all the stations.

Thanks to the information saved by the AP during the last transmission period, regarding the rates  $r^i$  used to transmit the frames  $f_i$ , the rate adaptation algorithm estimates the joint reception probabilities  $P_{r_j}$  associated with every available transmission rate  $r_j$  by counting how many multicast frames, transmitted with a given rate  $r_j$ , have been jointly received by all the stations of the multicast group. These values are used i) to compute the transmission rate  $r_b(e + 1)$  for the next super-frame, given the initial choice  $r_b(1) = 9Mb/s$ ; and ii) to choose the rate  $r_l$  that will be used to transmit the next *look around* frame. To this end, for each rate the algorithm maintains the set of values described in Table 1. Table 2 summarizes all the used parameters.

Table 1: Information maintained by the AP for each rate  $r_j$

Name	Description
$np_{r_j}$	Number of multicast data frames transmitted with rate $r_j$ since the last time when $P_{r_j}$ has been computed. This value is reset to 0 every time $P_{r_j}$ is recomputed
$nj_{r_j}$	Number of multicast data frames transmitted with rate $r_j$ and jointly received by all STAs of the multicast group since $P_{r_j}$ was recomputed. Reset to 0 every time $P_{r_j}$ is recomputed
$ls_{r_j}$	Sequence number of the last multicast data frame transmitted with rate $r_j$
$P_{r_j}$	Joint reception probability of a multicast data frame transmitted with rate $r_j$ . Recomputed every time $np_{r_j} \geq \beta$ (see Table 2)
$S_{r_j}$	Measure of successfulness of a multicast frame transmitted with rate $r_j$ , recomputed with $P_{r_j}$

In the following we assume that the feedback protocol completed successfully and the AP received a feedback from every station in the multicast group. The algorithm updates the value of the joint reception probability  $P_{r_j}$  for every rate  $r_j$  such that  $np_{r_j} \geq \beta$  using the following rule:

$$P_{r_j} \leftarrow (1 - \lambda)P_{r_j} + \lambda (nj_{r_j}/np_{r_j}) \quad (1)$$

In Eq. 1,  $\lambda \in (0, 1)$  is the EWMA (Exponentially Weighted Moving Average) weight and for every transmission rate  $r_j$  the corresponding  $P_{r_j}$  is initially set to 0.

For every rate  $r_j$ , whose corresponding joint reception probability value ( $P_{r_j}$ ) has been recomputed after the execution of the feedback protocol, the algorithm recomputes the measure of successfulness  $S_{r_j}$  that is a function  $f(*)$  of the joint reception probability

Table 2: Parameters used by the proposed rate adaptation algorithm

Name	Description
$\gamma$	Percentage of <i>look around</i> frames in a super-frame, $\gamma \in (0, 1)$ . <i>Look around</i> frames transmitted with suboptimal rate may be lost: for this reason $\gamma$ should be low ( <i>default: 0.1</i> )
$\beta$	Before $P_{r_j}$ is recomputed, $np_{r_j}$ must reach minimum $\beta \geq 1$ . High values may lead to better estimation of the joint reception probability, reducing oscillations in the choice of the optimal rate. Nevertheless, high values can decrease the algorithm's convergence time ( <i>default: 10</i> )
$\alpha$	Threshold value of term $C_t$ in Eq. 5. It is used to assure every transmission rate $r_l$ has a chance to be chosen to transmit the next <i>look around</i> frame ( <i>default: 0.05</i> )
$\lambda$	EWMA parameter, $\lambda \in [0, 1]$ : adapts convergency to new channel conditions ( <i>default: 0.7</i> )
$\sigma_1, \sigma_2, \sigma_3$	Weights of term $A_t, B_t$ and $C_t$ in Eq. 5 ( <i>default: 1, 0.2, 5</i> )
$N$	Number of multicast frames in a super-frame ( <i>default: 128</i> )
$i$	Multicast frame sequence number: starts with 1 and incremented after every transmission
$x$	Loss threshold value. Used to limit frame loss probability to $(1 - x)$ ( <i>default: 0.04</i> )

$P_{r_j}$ . We considered two different cases for  $f(*)$ : in the first case  $S_{r_j}$  is given by  $P_{r_j}$  multiplied by the bit rate of the rate  $r_j$  and can be expressed as  $S_{r_j} = f(P_{r_j}) = P_{r_j} \cdot r_j$ .

In this case the algorithm sets the transmission rate  $r_b(e + 1)$  to be used in the next transmission period to the highest rate  $r_j$  whose measure of successfulness is maximum among all the available transmission rates:

$$r_b(e + 1) \leftarrow \max_{r_j} \left\{ r_j \mid S_{r_j} = \max_{r_k} S_{r_k} \right\} \quad (2)$$

In the second case we set  $S_{r_j} = P_{r_j}$  and the algorithm chooses the highest rate  $r_j$  such that  $S_{r_j} \geq x$ , where  $x$  is a fixed threshold value:

$$r_b(e + 1) \leftarrow \max_{r_j} \left\{ r_j \mid S_{r_j} \geq x, 0 < x < 1 \right\} \quad (3)$$

Note that in the first case the algorithm maximizes the throughput, in the second case, instead, the algorithm limits the probability to lose a frames to  $(1 - x)$ .

Regarding the *look around* frames, a rate  $r_l \neq r_b$  is chosen with a probability  $p_{r_l}^{la}$  that is influenced by the current joint reception probability ( $P_{r_l}$ ), by the number of multicast frames that have been sent since the last multicast frames transmitted with that rate ( $i - ls_{r_l}$ ) and by the total number of multicast frames transmitted with that rate since  $P_{r_l}$  has been recomputed ( $np_{r_l}$ ). On the one hand, each rate  $r_l$  must have a positive probability  $p_{r_l}^{la}$  to be chosen for the next *look around* frame (we want to continuously evaluate channel quality for all available rates); on the other hand, we do not want to transmit too many multicast frames with a rate that has shown bad performance to limit frames losses. Eq. 4 and 5 illustrates how the AP computes  $p_{r_l}^{la}$ :

$$p_{r_l}^{la} = W_{r_l} / \sum_{k(\neq i)} W_{r_k} \quad (4)$$

$$W_{r_t} = \sigma_1 A_t + \sigma_2 B_t + \sigma_3 C_t \quad (5)$$

where parameters  $\sigma_1$ ,  $\sigma_2$  and  $\sigma_3$  in Eq.5 weigh  $A_t$ ,  $B_t$  and  $C_t$ , respectively:

$$A_t = \begin{cases} \frac{\beta - np_{r_t}}{\beta} & \beta - np_{r_t} \geq 0 \\ 0 & otherwise \end{cases} \quad (6)$$

$$B_t = \frac{i - ls_{r_t}}{\max_{r_k}(i - ls_{r_k})} \quad (7)$$

$$C_t = \frac{P_{r_t} + \alpha}{\sum_{h(\neq t)} (P_{r_h} + \alpha)} \quad (8)$$

Term  $A_t$  states that the probability to choose rate  $r_t$  for the next *look around* frame must decrease when the total number of multicast frames transmitted at that rate since  $P_{r_t}$  has been computed is close or equal to  $\beta$ , where the latter is the minimum value that must be reached by  $np_{r_t}$  before  $P_{r_t}$  can be recomputed. Term  $B_t$  expresses the fact that the probability to choose rate  $r_t$  must be higher for rates that have not been tested for a long time. Term  $C_t$  states that the probability to choose rate  $r_t$  must be higher for rates that have shown good performance until now. Parameter  $\alpha$  is used as a threshold value so that all rates have  $C_t > 0$ .

## 5. Numerical Results

This section illustrates the experimental results achieved by the proposed multicast rate adaptation algorithm. We compare the following four algorithms:

1. *Best throughput*: this algorithm tries to maximize the throughput; the rate for the next super-frame is determined by Eq. 2;
2. *Limited losses*: this algorithm tries to limit losses; the rate for the next super-frame is determined by Eq. 3;
3. *Fixed rate*: this algorithm<sup>4</sup> is used by most of the commercial wifi cards; all multicast frames are transmitted at the lowest rate, 6Mb/s for IEEE 802.11g;
4. *Linear increase/Multiplicative decrease*: this algorithm, proposed in [10], uses the same super-frame format and the same *feedback* protocol described in Section 3.1: the rate for the next super-frame is chosen comparing the expected transmission times of two previous transmission periods according to the following rule:

$$j = \text{index\_of}(\mathbf{R}, r_b(e)), \quad (9)$$

$$r_b(e+1) = \begin{cases} \mathbf{R}[j+1] & \frac{T(e)}{T(e-1)} \leq 1 \\ \mathbf{R}[j-2] & \frac{T(e)}{T(e-1)} > 1. \end{cases} \quad (10)$$

where  $\mathbf{R}$  represents the list of available transmission rates, and Eq. 9 returns the index of rate  $r_b(e)$  within  $\mathbf{R}$ . Assuming the same size  $L$  for all multicast frames, the expected transmission time of a frame is  $T(e) = L/(P(e) \cdot r_b(e))$ , where  $P(e)$  is joint reception probability computed considering only the  $N$  multicast frames transmitted during the transmission period  $e$  (EWMA is not performed).

<sup>4</sup>This is the only one of the four tested algorithms that doesn't work on top of the WMP.

## 5.1 Experimental Methodology

To compare the proposed mechanisms, we used the testbed showed in Figure 5, located inside the Department of Information Engineering at the University of Brescia. We positioned the AP (black circle) inside the LAB together with nine fixed stations (black squares) lying from 2 to 8 meters away the AP. We used a battery-powered Alix 2D2 as mobile node. All nodes are equipped with Broadcom 4318 wireless NICs (802.11g compliant) and run the WMP. We tested separately each of the four considered systems

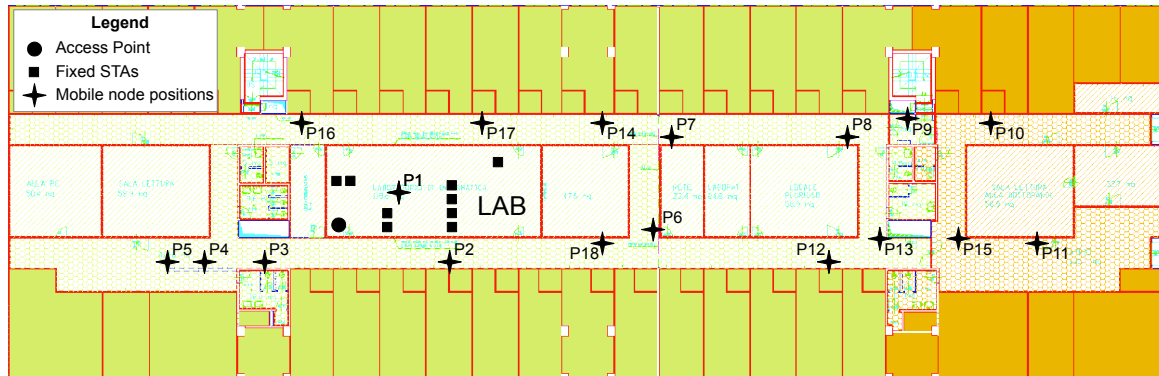


Figure 5: Testbed with one AP, nine fixed STAs and one mobile node.

with the mobile station located in 18 different positions, ranging from a 3m minimum distance (P1) to a 55m maximum distance (P11) with respect to the AP. However, we point out that the delivery probability is not strictly correlated to the distance of the mobile node from the AP; instead, it's the massive presence of floor-to-ceiling walls, solid steel glass doors and other types of obstacles that makes some positions more adverse than others to the propagation of the signal. In each test, that we repeated five times, the AP greedily transmitted 100000 multicast frames: we ended up with  $18(\text{positions}) \times 4(\text{systems}) \times 5(\text{repetitions})$ . We fixed packet payload to 1470 bytes similarly to many video streaming applications based on the Real Time Protocol (RTP).

We set the parameters used by the proposed algorithms to the *default* values indicated in Table 2: their choice came from preliminary experimental observations that followed a quick analysis on the influence of each parameter on algorithms' behavior.

## 5.2 Network Performance Analysis

We consider as performance metrics the *average goodput*, namely the average bandwidth actually used for succesful transmission of useful application layer data, the *average delay* and the *average losses* measured by the ten STAs.

Figure 6a shows the average goodput, measured by the ten STAs, as a function of the mobile node position: the *best throughput* version of the proposed algorithm outperforms the others in all experimental scenarios. In particular, we get a performance increase ranging from 102% to 250% with respect to the standard *fixed rate* solution, from 15% to 33% with respect to the *linear-increase/multiplicative-decrease* one, and from 15% to 97% if compared with the *limited losses* version of the algorithm. Though the latter performs slightly worse than the *linear-increase/multiplicative-decrease*, we will see shortly that it provides better performance in terms of losses.

Figure 6b shows the average one-way transmission delay. Results are specular wrt the goodput: the standard *fixed rate* solution exhibits always higher delays than the



*linear-increase/multiplicative-decrease* and the *limited losses* solutions; whereas the *best throughput* algorithm achieves the lowest delay. Quantitatively the last three algorithms reduce the one-way delay up to 5 times with respect to the *fixed rate* solution.

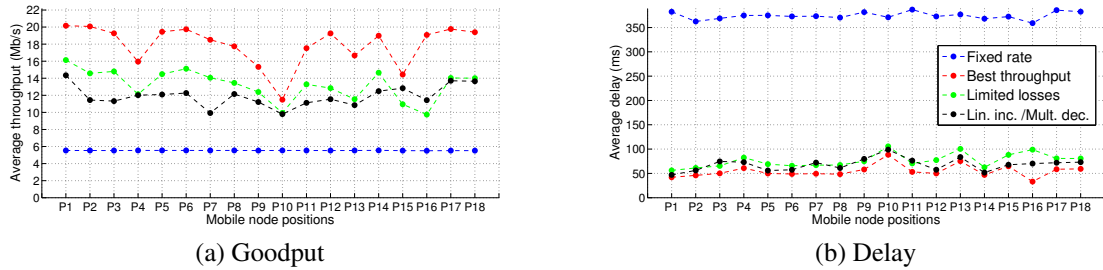


Figure 6: Average goodput and delay as a function of the mobile node's position.

Finally, frame losses are illustrated in Figure 7. Dashed lines represent the average losses experienced by all the stations, whereas crosses correspond to the losses of the single mobile node. Even though the *linear-increase/multiplicative-decrease* algorithm provides good performance in terms of goodput and delay, the high number of frame losses may affect the Quality of Service of multimedia streams.

The *limited losses* algorithm well approaches the performance achieved by the *fixed rate* scheme, which represents the best solution. It can be further observed that the *best throughput* mechanism causes a slightly higher number of frame losses than the *limited losses*. Indeed, when the mobile node is in a critical position the *limited losses* algorithm sacrifices the goodput to maintain losses under 4% for all the STAs; whereas the *best throughput* version selects greedily the transmission rate that maximizes the throughput, thus leading the mobile node to lose up to 10% of the transmitted frames.

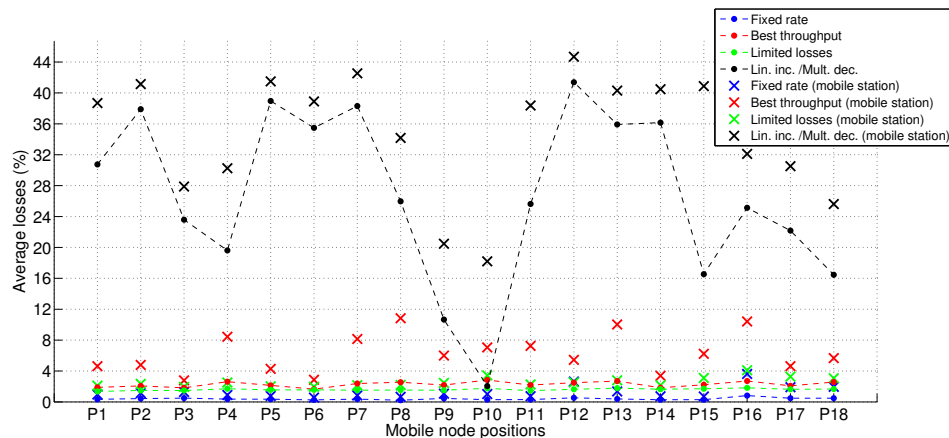


Figure 7: Average losses as a function of the mobile nodes's position.

## 6. Conclusion

We proposed in this paper an innovative feedback protocol to estimate the reception probability correlation of the wireless links connecting an AP to a group of STAs. We implemented the protocol on the Wireless MAC Processor (WMP) architecture in order to fully exploit the flexibility provided by current wireless cards and provide a set of standard functionalities to rapidly develop new rate adaptation schemes.

On top of the protocol we developed three algorithms to select adaptively the transmission rate of multicast data frames and we tested them on a real-life testbed to validate our modular implementation.

Specifically, our solutions increase the overall performance in terms of throughput and transmission delay, with losses approaching the same values achieved by the fixed rate scheme; thus representing a viable solution to increase the number of multicast multimedia streams that can be transmitted using the 802.11 technology.

## Acknowledgment

This work was supported by the European Commission through the FP7 FLAVIA Project.

## References

- [1] E. H. Ong, J. Knecht, O. Alanen, Z. Chang, T. Huovinen, and T. Nihtila. Ieee 802.11ac: Enhancements for very high throughput wlans. In *Proceedings of the 22nd IEEE symposium on Personal Indoor and Mobile Radio Communications*, pages 849–853, 2011.
- [2] Mac enhancements for robust audio video streaming. *IEEE Std 802.11aa-2012*, pages 1–162, 2012.
- [3] I. Tinnirello, G. Bianchi, P. Gallo, D. Garlisi, F. Giuliano, and F. Gringoli. Wireless mac processors: Programming mac protocols on commodity hardware. *Proc. of IEEE INFOCOM*, pages 1269–1277, 2012.
- [4] A. Kamerman and L. Monteban. Wavelan-ii: a high-performance wireless lan for the unlicensed band. *Bell Labs Technical Journal*, 2:118–133, 1997.
- [5] J. Kim, S. Kim, S. Choi, and D. Qiao. Cara: Collision-aware rate adaptation for ieee 802.11 wlans. *Proc. of IEEE INFOCOM*, pages 1–11, 2006.
- [6] A. Basalamah, H Sugimoto, and T. Sato. Rate adaptive reliable multicast mac protocol for wlans. *Proc. of IEEE Vehicular Technology Conference*, pages 1216–1220, 2006.
- [7] M. Vutukuru, H. Balakrishnan, and K. Jamieson. Cross-layer wireless bit rate adaptation. *Proc. of ACM SIGCOMM*, 39(4):3–14, 2009.
- [8] J. Villalon, P. Cuenca, L. Orozco-Barbosa, Y. Seok, and T. Turletti. Arsm: a cross-layer auto rate selection multicast mechanism for multi-rate wireless lans. *Communications, IET*, 1(5):893 –902, oct. 2007.
- [9] A. Basalamah and T. Sato. Adaptive fec reliable multicast mac protocol for wlan. In *Vehicular Technology Conference, 2007. VTC-2007 Fall. 2007 IEEE 66th*, pages 244–248, 30 2007-oct. 3 2007.
- [10] F. Gringoli A. Capone S. Paris, N. Facchi. An innovative rate adaptation algorithm for multicast transmissions in wireless lans. *IEEE 77th Vehicular Technology Conference (VTC2013-Spring)*, 2-5 June 2013.